INTELIGENCIA DE NEGOCIO

2019 - 2020

- Tema 1. Introducción a la Inteligencia de Negocio
- Tema 2. Minería de Datos. Ciencia de Datos
- Tema 3. Modelos de Predicción: Clasificación, regresión y series temporales
- Tema 4. Preparación de Datos
- Tema 5. Modelos de Agrupamiento o Segmentación
- Tema 6. Modelos de Asociación
- Tema 7. Modelos Avanzados de Minería de Datos.
- Tema 8. Big Data

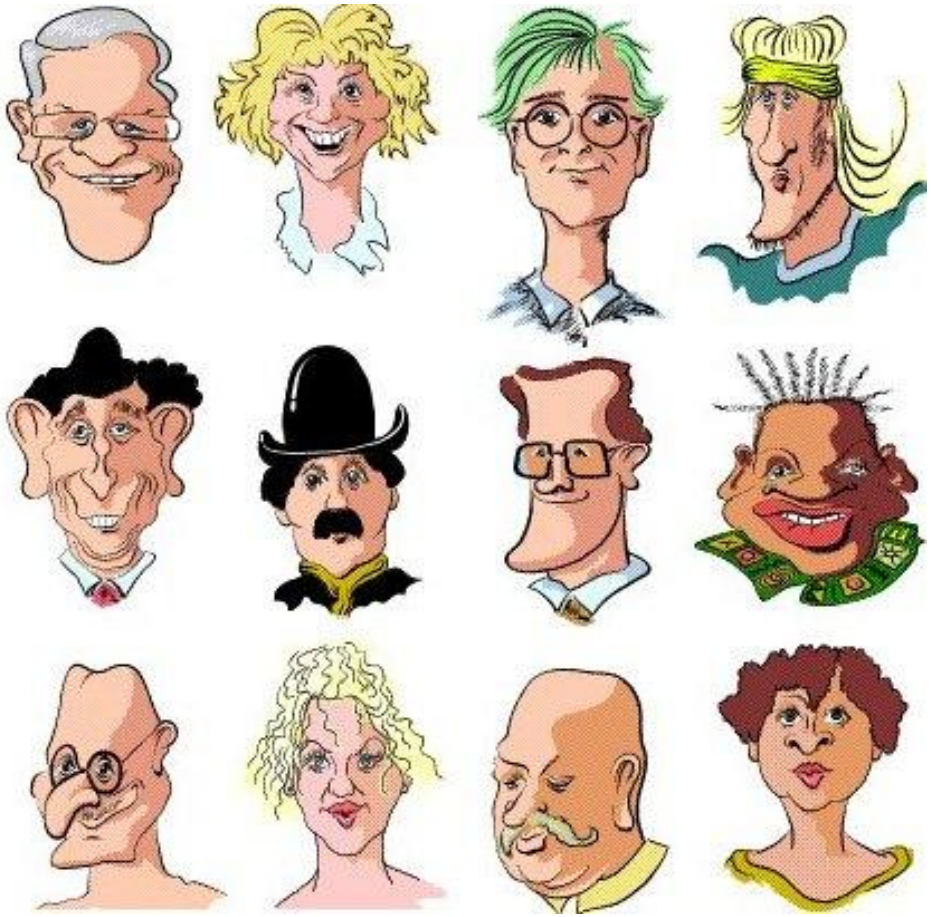
Objetivos

- Entender qué es un problema de agrupamiento (clustering)
- Conocer los criterios que se utilizan para evaluar un algoritmo de agrupamiento
- Entender el modo de funcionamiento de algunos algoritmos de agrupamiento.

Motivación

Hay problemas en los que deseamos agrupar las instancias creando clusters de similares características

Ej. Segmentación de clientes de una empresa



Contenidos sobre Clustering

1. **Clustering/agrupamiento/segmentación**
2. Medidas de distancia y similaridad
3. Distintas aproximaciones al clustering
4. Métodos basados en particionamiento
5. Métodos jerárquicos

1.1. Definición de *clustering*

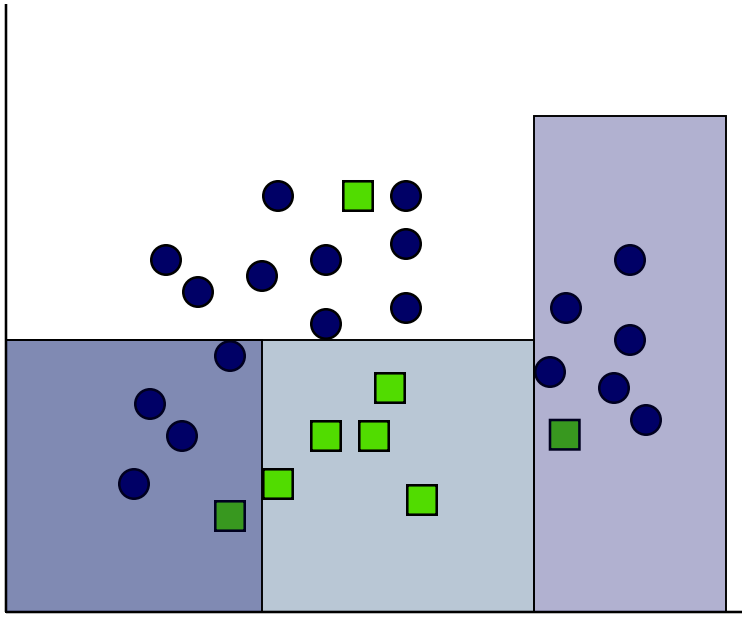
- *Cluster:* un grupo o conjunto de objetos
  - **Similares** a cualquier otro incluido en el mismo grupo
  - **Distintos** a los objetos incluidos en otros grupos
- *Clustering* (análisis *cluster*):
  - Segmentar una población heterogénea en un número de subgrupos homogéneos o *clusters*
- *Clustering* puede verse como clasificación no supervisada, las clases no están predefinidas
- Aplicaciones típicas:
  - Como una tarea de preprocesamiento antes de aplicar otra técnica de descubrimiento del conocimiento
  - Como técnica de descubrimiento del conocimiento para obtener información acerca de la distribución de los datos (p.e.: encontrar clientes con hábitos de compra similares)

1.1. Definición de *clustering*

Cuando se aplican algoritmos de *clustering* a problemas reales, nos enfrentamos a:

- Dificultad en el manejo de *outliers*
  - Se pueden ver como *clusters* solitarios
  - Se puede forzar a que estén integrados en algún *cluster* → suele implicar que la calidad de los *clusters* obtenidos es baja
- Si se realiza en BBDD dinámicas implica que la pertenencia a *clusters* varía en el tiempo
  - Los resultados del *clustering* son dinámicos
- Interpretar el significado de cada *cluster* puede ser difícil
- No hay una única solución para un problema de *clustering*. No es fácil determinar el número de *clusters*

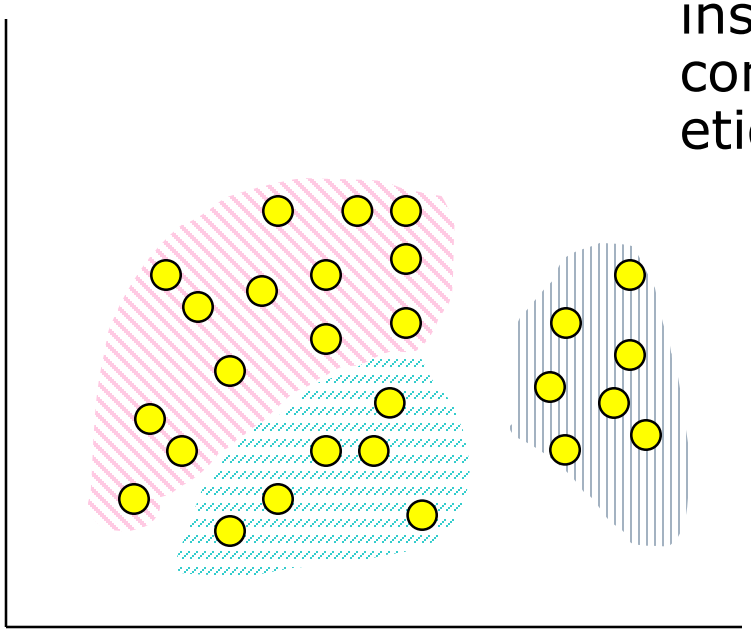
1.2. Clustering vs. clasificación



Clasificación: Aprendizaje supervisado.

Aprende, a partir de un conjunto de instancias pre-etiquetadas, un método para predecir la clase a que pertenece una nueva instancia

1.2. Clustering vs. clasificación



Aprendizaje no supervisado:

Encuentra un agrupamiento de instancias “natural” dado un conjunto de instancias no etiquetadas

1.3. Aplicaciones

- Marketing: descubrimiento de distintos grupos de clientes en la BD. Usar este conocimiento en la política publicitaria, ofertas, …
- Uso de la tierra: Identificación de áreas de uso similar a partir de BD con observaciones de la tierra (cultivos, …)
- Seguros: Identificar grupos de asegurados con características parecidas (siniestros, posesiones, ….). Ofertarles productos que otros clientes de ese grupo ya poseen y ellos no
- Planificación urbana: Identificar grupos de viviendas de acuerdo a su tipo, valor o situación geográfica
- WWW: Clasificación de documentos, analizar ficheros .log para descubrir patrones de acceso similares, …

1.4. Bondad de un análisis *cluster*

- Un buen método de *clustering* debe producir *clusters* en los que:
  - Se maximize la similaridad *intra-cluster*
  - Se minimize la similaridad *inter-cluster*
- La calidad del *clustering* resultante depende tanto de la medida de similaridad utilizada como de su implementación
- Medidas de similaridad/disimilaridad: normalmente una función de distancia: d(i, j)
- Las funciones de distancia son muy sensibles al tipo de variables usadas, así su definición puede cambiar según el tipo: medidas por intervalos, booleanas, categóricas (nominales), ordinales, …
- Es posible dar peso a ciertas variables dependiendo de distintos criterios (relativos a su aplicación, …)
- En general, es complicado dar definiciones para términos como "suficientemente similar", así que algunas respuestas serán subjetivas y dependientes de umbrales

1.5. Propiedades deseables en un método de *clustering* en minería de datos

- Escalables
- Capacidad para tratar distintos tipos de variables
- Capacidad para descubrir *clusters* con formas arbitrarias
- Requisitos mínimos de conocimiento del dominio para determinar los parámetros de entrada
- Capacidad para tratar datos con ruido y *outliers*
- Insensible al orden de los registros de entrada
- Capacidad para incorporar restricciones del usuario
- Válido para registros de alta dimensionalidad
- Resultados interpretables

Contenidos sobre Clustering

1. Clustering/agrupamiento/segmentación
2. **Medidas de distancia y similaridad**
3. Distintas aproximaciones al clustering
4. Métodos basados en particionamiento
5. Métodos jerárquicos

2. Medidas de distancia y similaridad

- La propiedad más importante que debe verificar un *cluster* es que haya más cercanía entre las instancias que están dentro del *cluster* que respecto a las que están fuera del mismo (similitud entre instancias)

- ¿Qué es la similitud? ¿Cómo medir la similitud entre instancias?

(c) Eamonn Keogh, eamonn@cs.ucr.edu

2. Medidas de distancia y similaridad

- La definición de la medida de distancia depende normalmente del tipo de variable:
  - Variables continuas
  - Variables binarias/boolenas
  - Variables nominales/categóricas
  - Variables ordinales
  - Variables mixtas

2. Medidas de distancia y similaridad

- El caso más simple: un único atributo numérico A
  Distancia(X,Y) = A(X) – A(Y)

- Varios atributos numéricos:
  - Distancia(X,Y) = Distancia euclídea entre X,Y

- Atributos nominales: La distancia se fija a 1 si los valores son diferentes, a 0 si son iguales

- Muchas medidas son sensibles al rango de cada variable, de modo que es necesario normalizarlas

- ¿Tienen todos los atributos la misma importancia?
  - Si no tienen igual importancia, será necesario ponderar los atributos

2. Medidas de distancia y similaridad

- Distancia de *Minkowski*:

$$d(i,j)=\sqrt[q]{(|x_{i_1}-x_{j_1}|^q+|x_{i_2}-x_{j_2}|^q+...+|x_{i_p}-x_{j_p}|^q)}$$

donde $i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ y $j = (x_{j1}, x_{j2}, \ldots, x_{jp})$ son dos objetos p-dimensionales, y q es un entero positivo

- Si $q = 1$, d es la distancia de Manhattan

$$d(i,j)=|x_{i_1}-x_{j_1}|+|x_{i_2}-x_{j_2}|+...+|x_{i_p}-x_{j_p}|$$

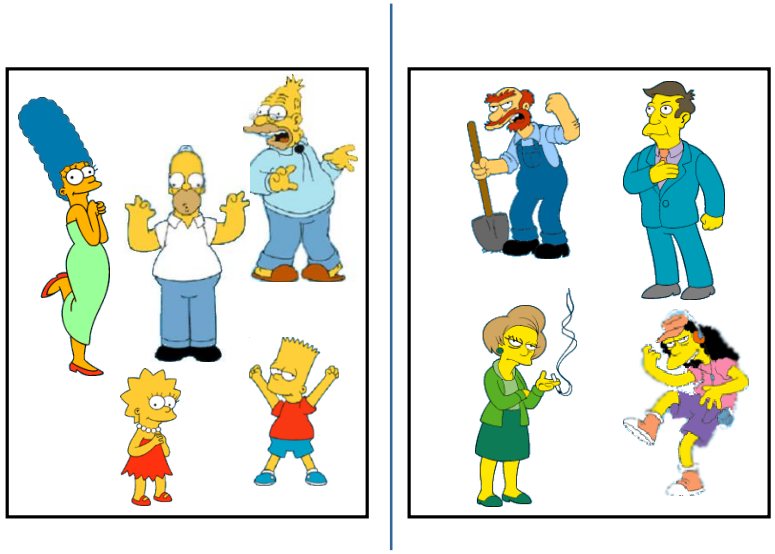
- *Si* $q = 2$, d es la distancia Euclídea
- Como ya hemos comentado se pueden usar pesos. Por ejemplo, Euclídea con pesos:

$$d(i,j)=\sqrt{w_1|x_{i_1}-x_{j_1}|^2+w_2|x_{i_2}-x_{j_2}|^2+...+w_p|x_{i_p}-x_{j_p}|^2}$$

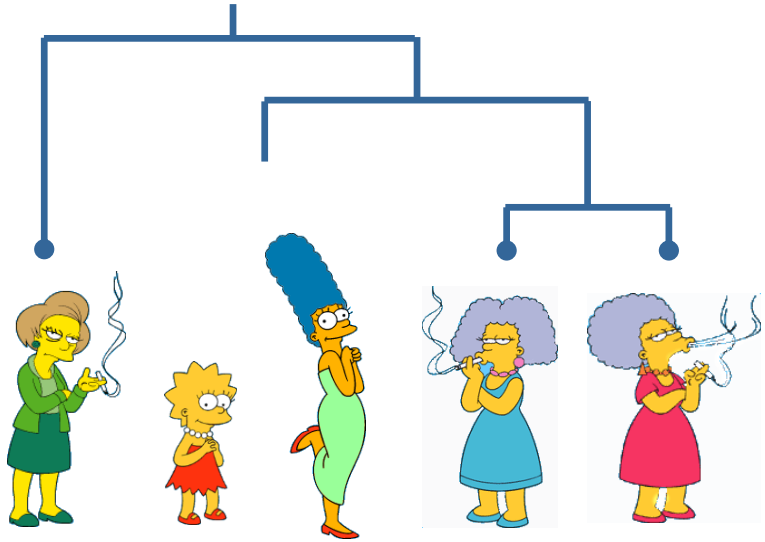
Contenidos sobre Clustering

1. Clustering/agrupamiento/segmentación
2. Medidas de distancia y similaridad
3. **Distintas aproximaciones al clustering**
4. Métodos basados en particionamiento
5. Métodos jerárquicos

Particional



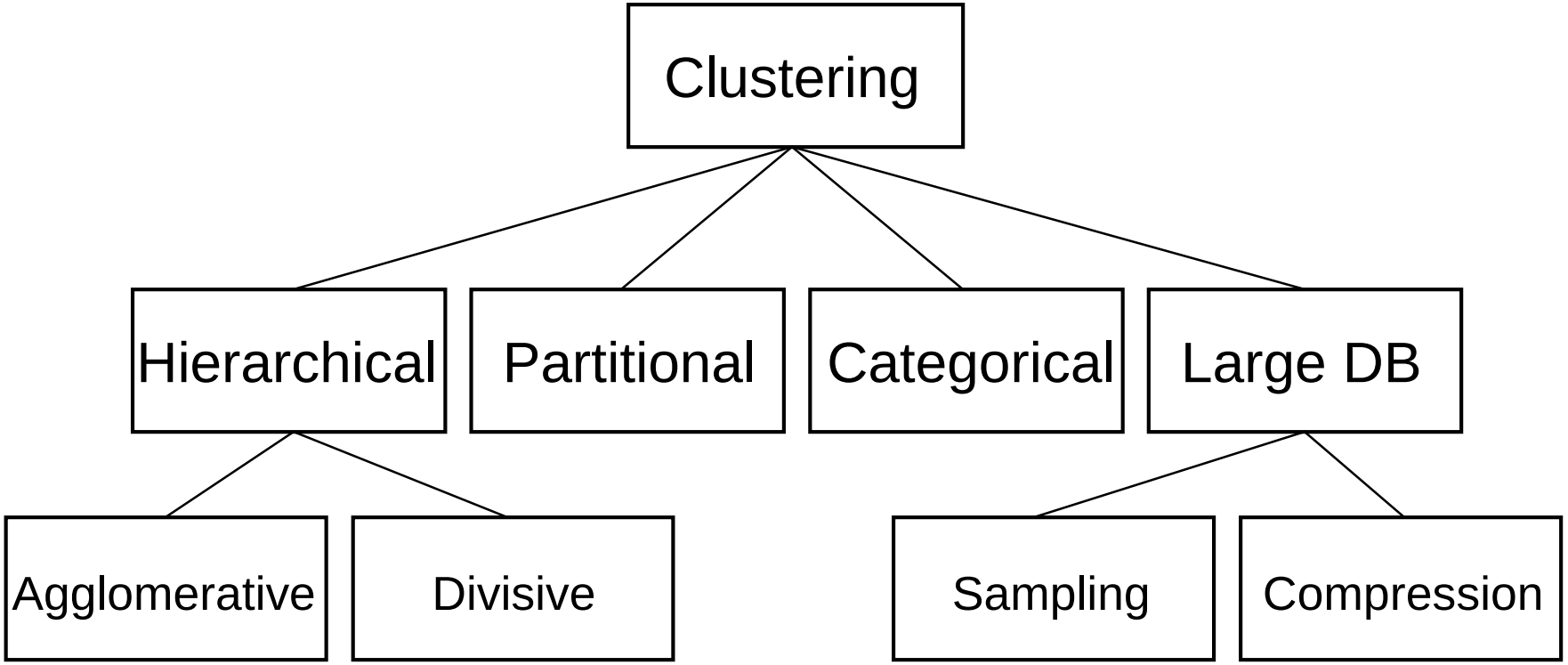
Jerárquico



3. Distintas aproximaciones al *clustering*

- Algoritmos de particionamiento: Construir distintas particiones y evaluarlas de acuerdo a algún criterio
- Algoritmos jerárquicos: Crear una descomposición jerárquica del conjunto de datos (objetos) usando algún criterio
- Otros:
  - Basados en densidad, utilizan funciones de conectividad y densidad
  - Basados en rejillas, utilizan una estructura de granularidad de múltiples niveles
  - Basados en modelos. Se supone un modelo para cada uno de los *clusters* y la idea es encontrar el modelo que mejor ajuste

3. Distintas aproximaciones al clustering



Contenidos sobre Clustering

1. Clustering/agrupamiento/segmentación
2. Medidas de distancia y similaridad
3. Distintas aproximaciones al *clustering*
4. **Métodos basados en particionamiento**
5. Métodos jerárquicos

4. Métodos basados en particionamiento

- Métodos basados en particionamiento: Construyen una partición de la base de datos *D* formada por *n* objetos en un conjunto de *k clusters*

- Dado un valor para *k*, encontrar la partición de D en *k clusters* que optimice el criterio de particionamiento elegido

- Métodos Heurísticos:
  - *k-means* (k medias): cada *cluster* se representa por el centro del *cluster*
  - *k-medoids* o PAM (particionamiento alrededor de los *medoides*): cada *cluster* se representa por uno de los objetos incluidos en el *cluster*

4. Métodos basados en particionamiento

Algoritmo *k-means*

- Necesita como parámetro de entrada el número de *clusters* deseado

- Es un algoritmo iterativo en el que las instancias se van moviendo entre *clusters* hasta que se alcanza el conjunto de *clusters* deseado

4. Métodos basados en particionamiento

Algoritmo *K-Means*

- Para k=1,...,K hacer
  - r[k] = punto seleccionado arbitrariamente de D

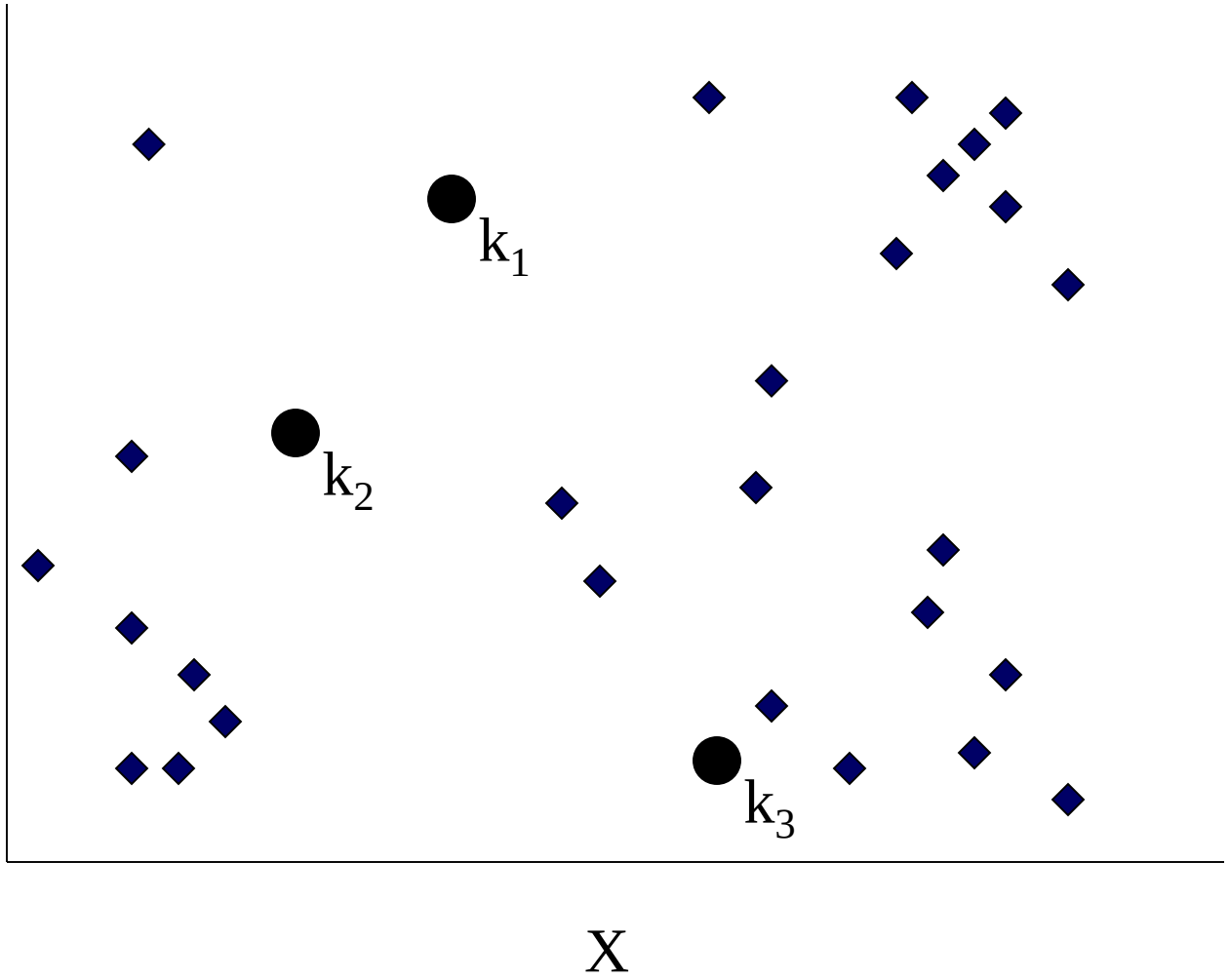
- Mientras haya cambios en los *clusters* $C_1,...,C_k$ hacer

  - Para k=1,...,K hacer // construir los *clusters*
    $C_k = \{ x \in D \mid d(r[k],x) \leq d(r[j],x)$
    para todo $j=1,...,K, \ j \neq k\}$

  - Para k=1,...,K hacer // calcular los nuevos centros
    r[k] = el punto medio de los objetos en C_k

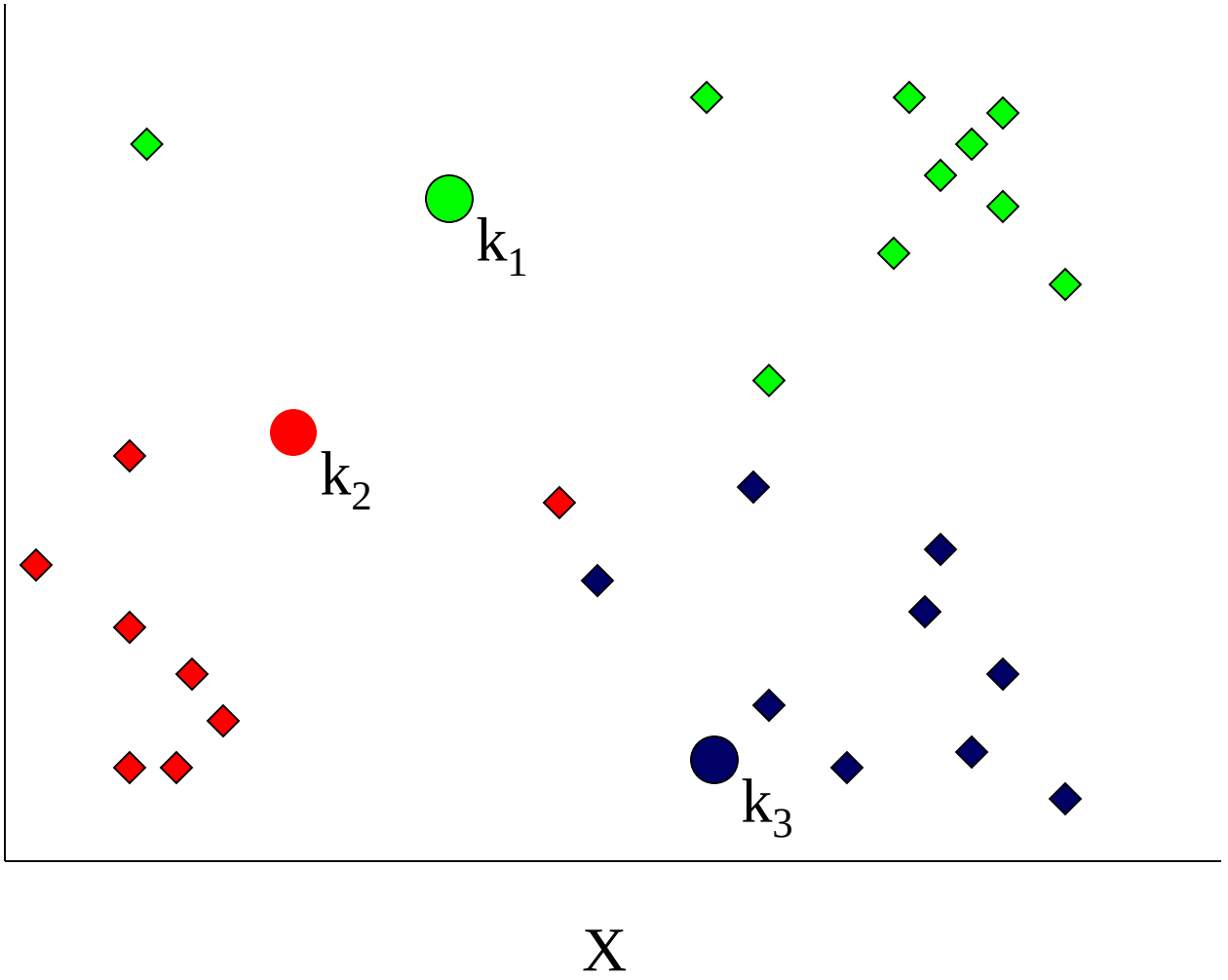
4. Métodos basados en particionamiento.
Ejemplo de *K-means*

Elige 3 centros de *clusters* (aleatoriamente)



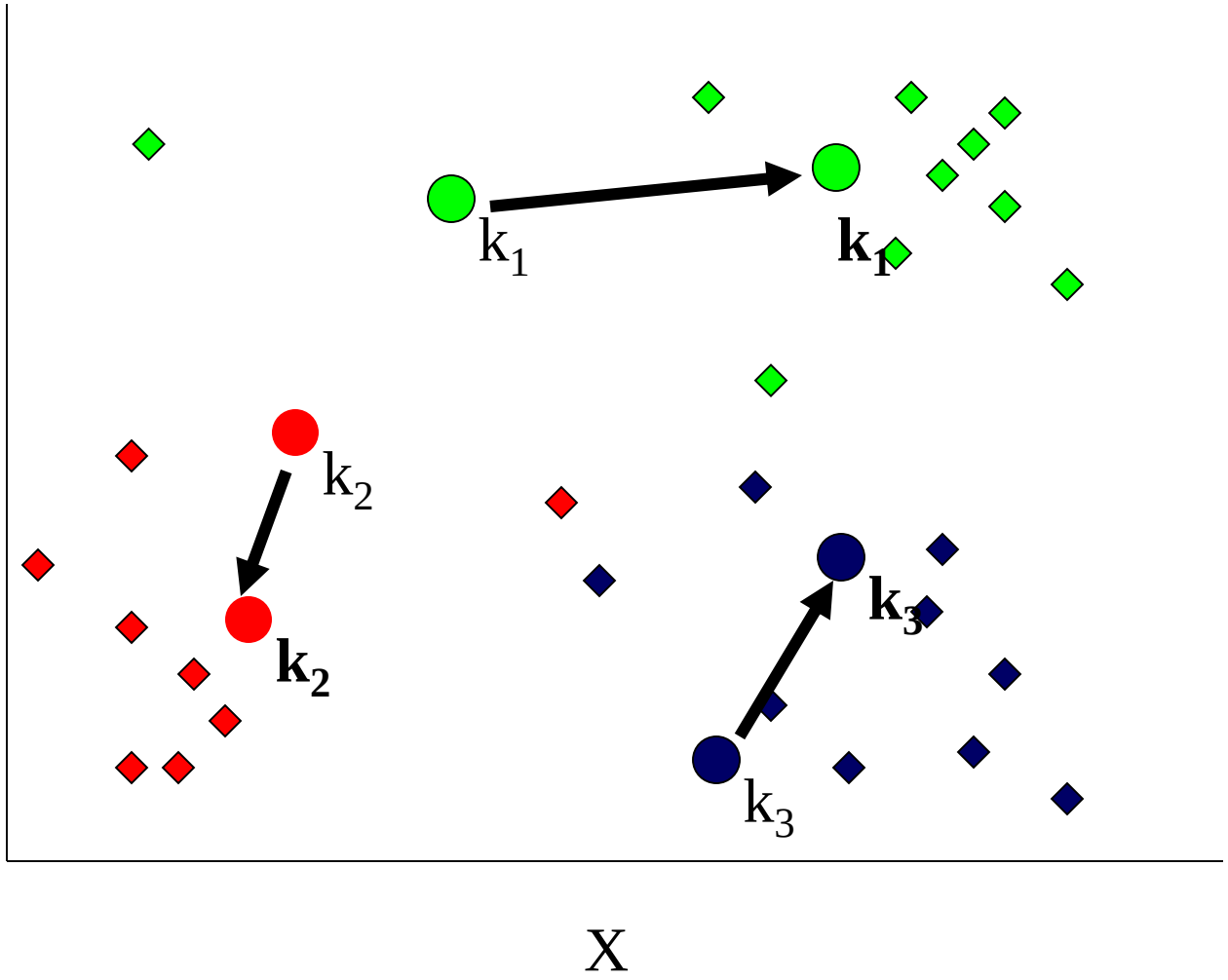
4. Métodos basados en particionamiento.
Ejemplo de *K-means*

Asigna cada punto al centro de *cluster* más cercano



4. Métodos basados en particionamiento.
Ejemplo de *K-means*

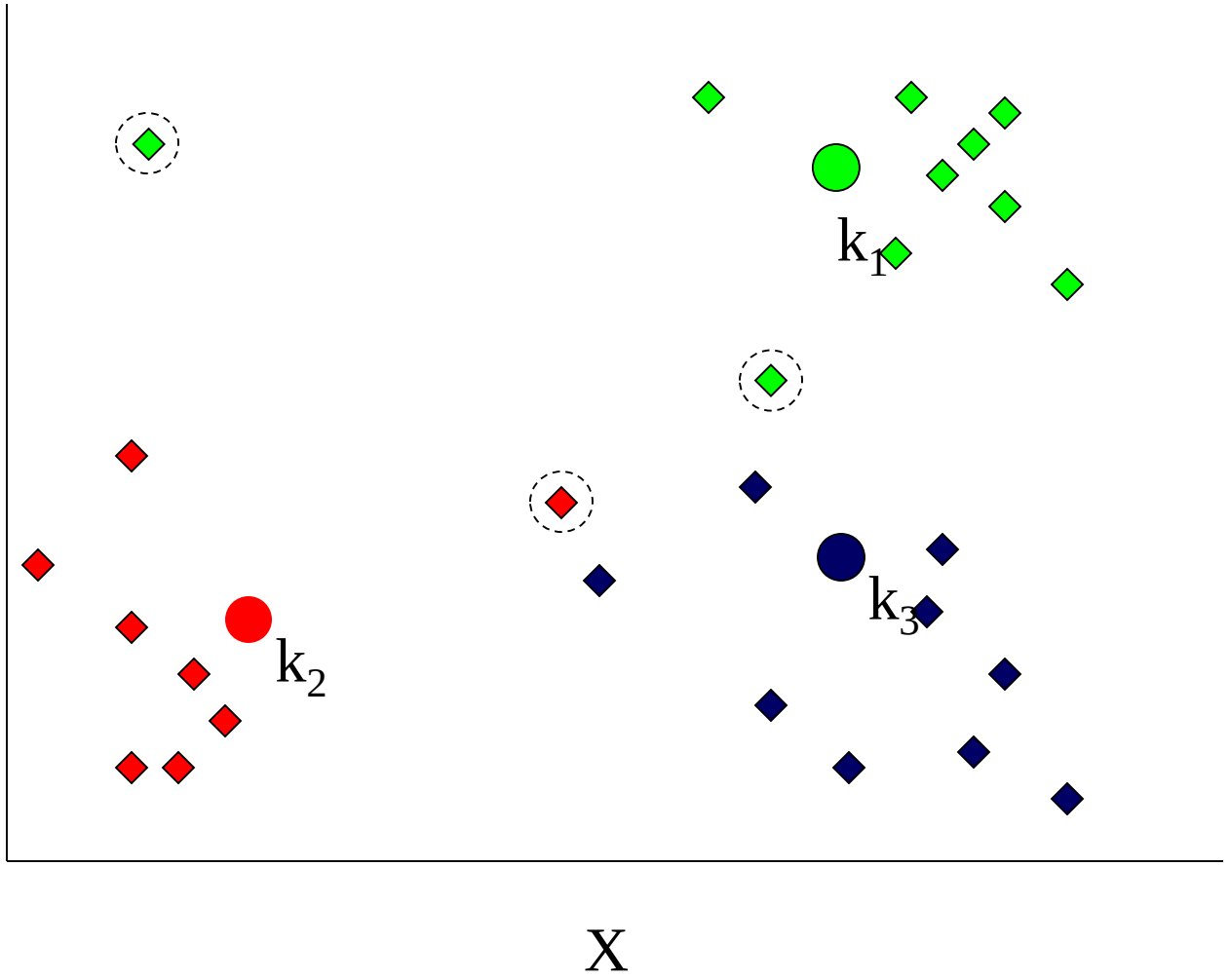
Mueve cada centro de *cluster* a la media de cada *cluster*



4. Métodos basados en particionamiento.
Ejemplo de *K-means*

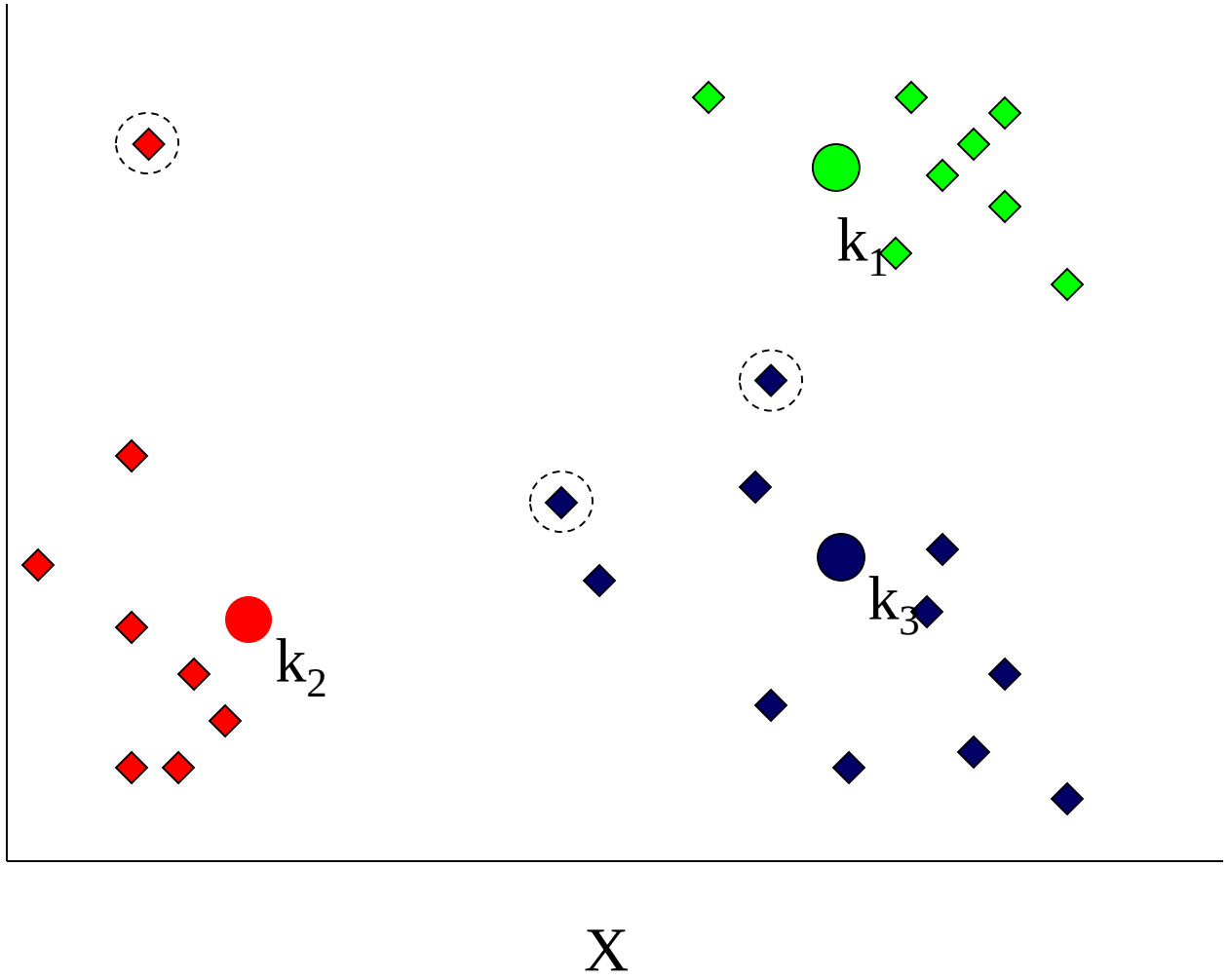
Reasigna los puntos más cercanos a diferentes centros de *clusters*

¿Qué puntos se reasignan?



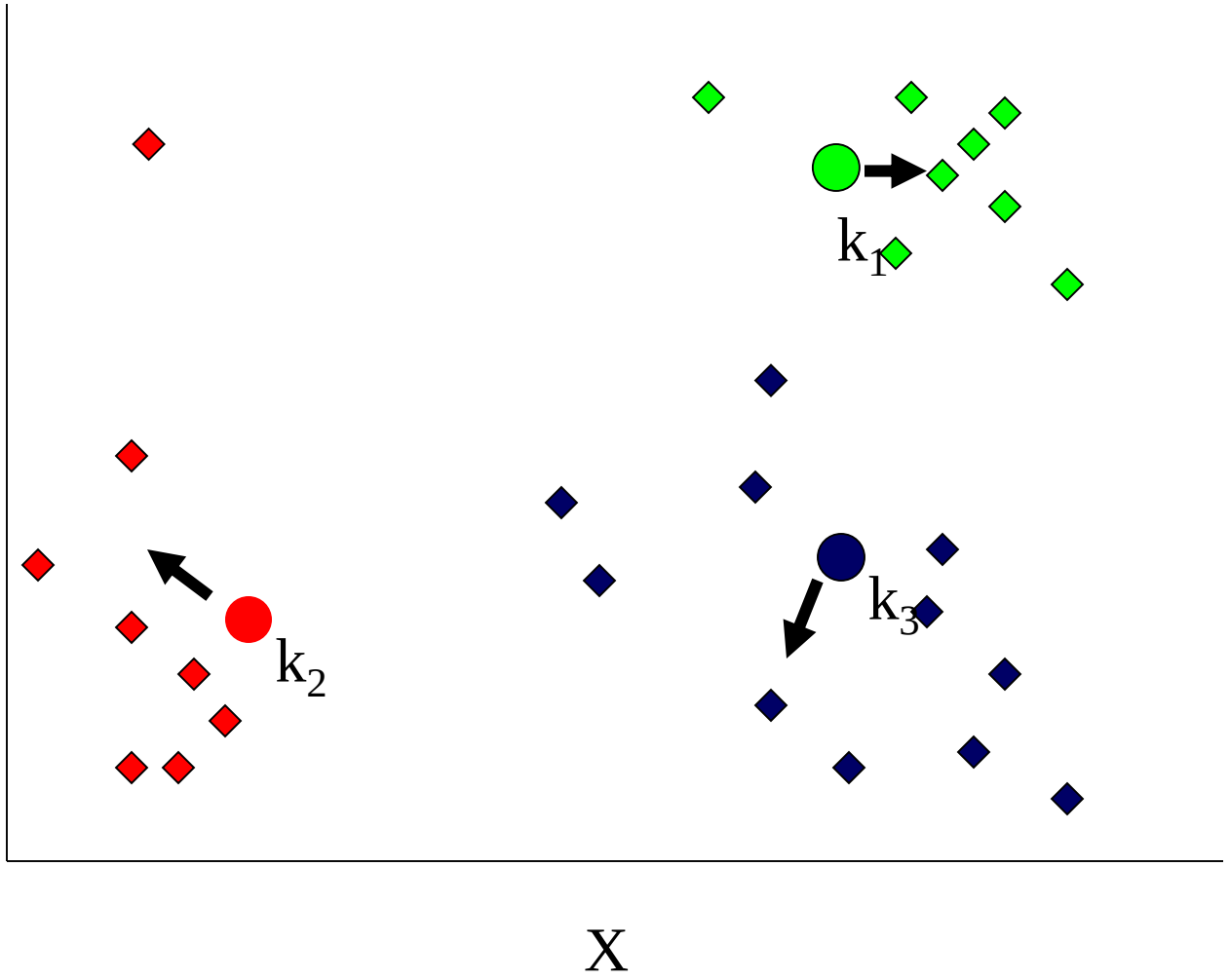
4. Métodos basados en particionamiento.
Ejemplo de *K-means*

Los tres puntos con movimiento



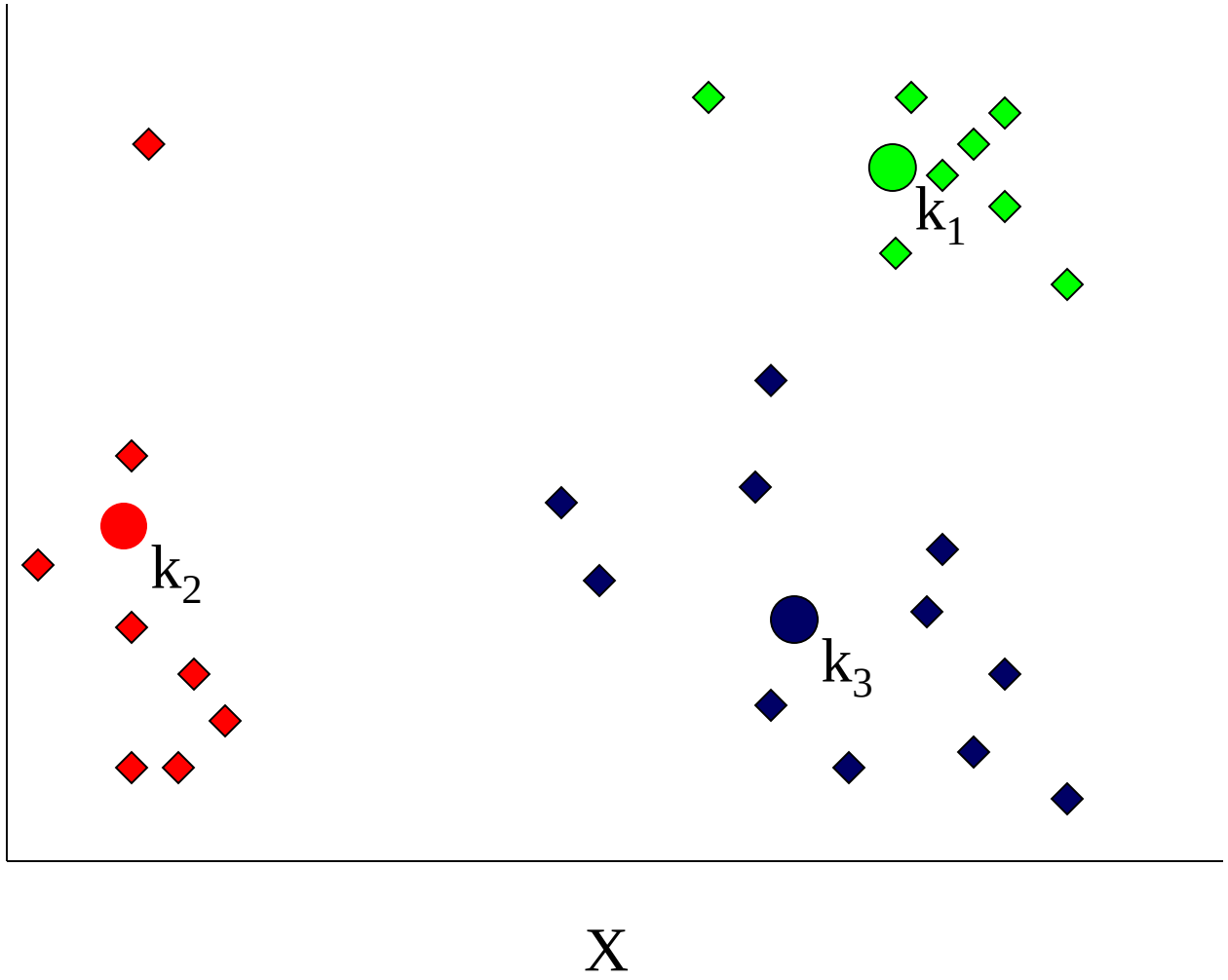
4. Métodos basados en particionamiento.
Ejemplo de *K-means*

Re-cálculo de los centros de *clusters*



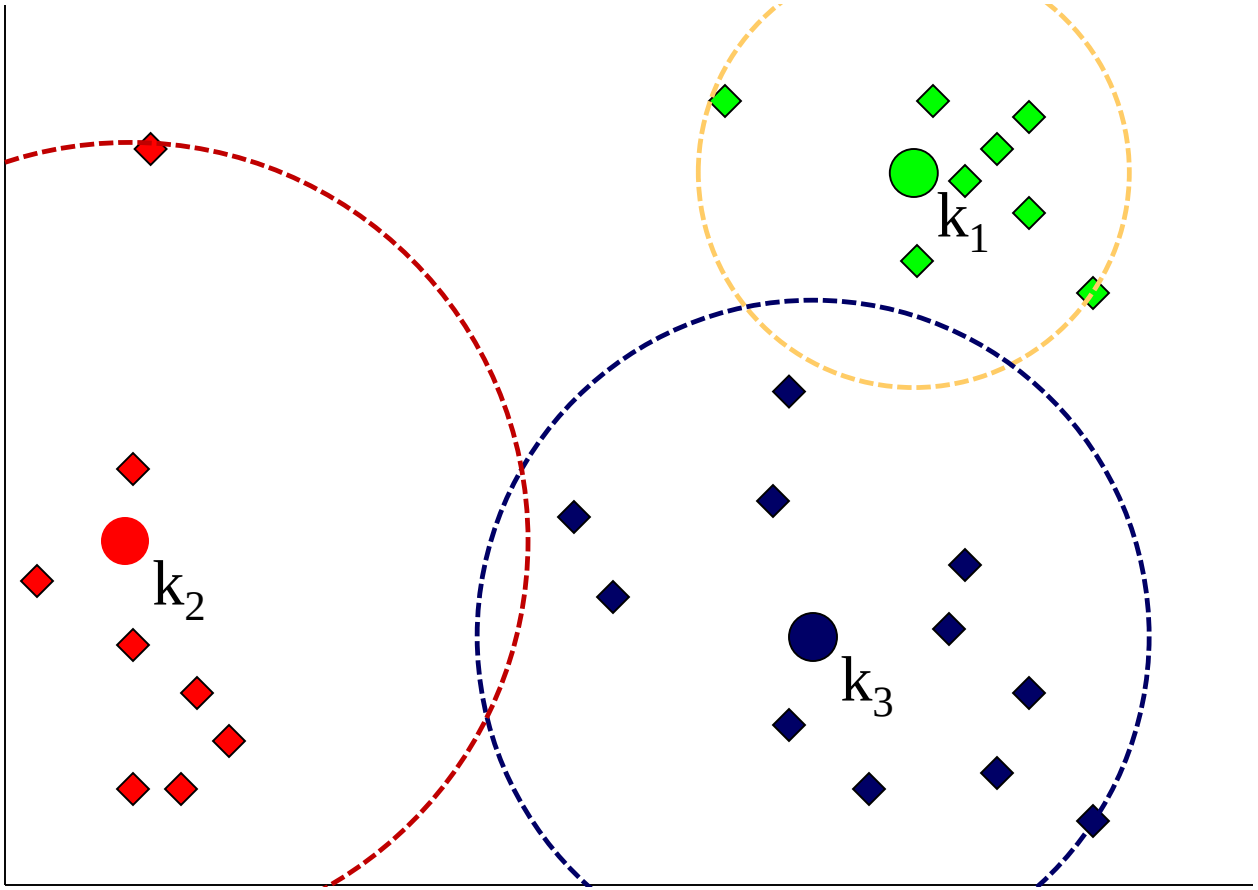
4. Métodos basados en particionamiento.
Ejemplo de *K-means*

Mover los centros a las medias de los *clusters*



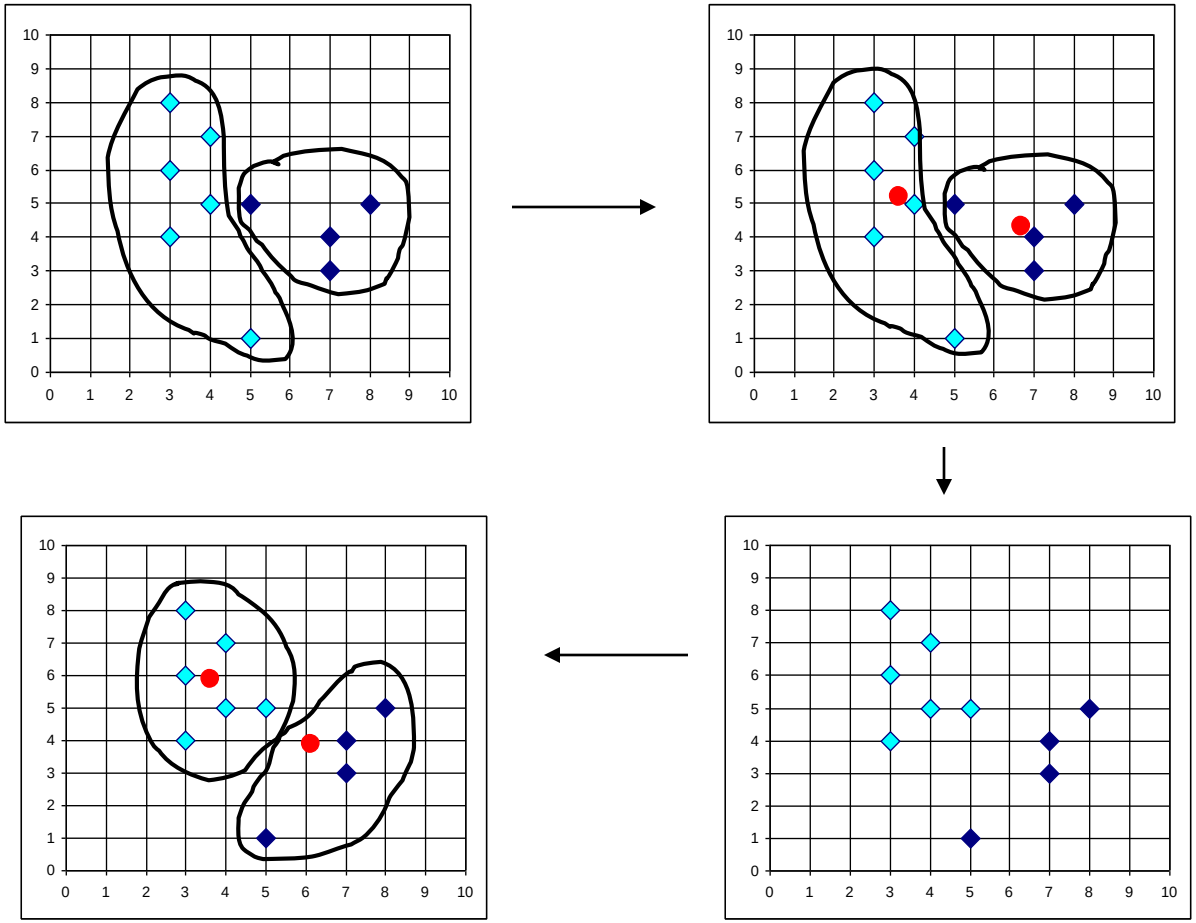
4. Métodos basados en particionamiento.
Ejemplo de *K-means*

Después de mover los centros, todos los puntos siguen perteneciendo a los mismos *clusters*, así que el proceso termina



4. Métodos basados en particionamiento. Ejemplo de *K-means*

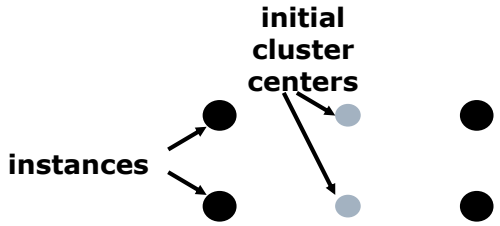
- Ejemplo:



4. Métodos basados en particionamiento. Algunos comentarios sobre k-means

Ventajas

- *Relativamente eficiente*: $O(tkn)$, donde n es # objetos, k es # *clusters*, y t es # iteraciones. Normalmente, $k, t << n$.
- Con frecuencia finaliza en un óptimo local, dependiendo de la elección inicial de los centros de *clusters*.



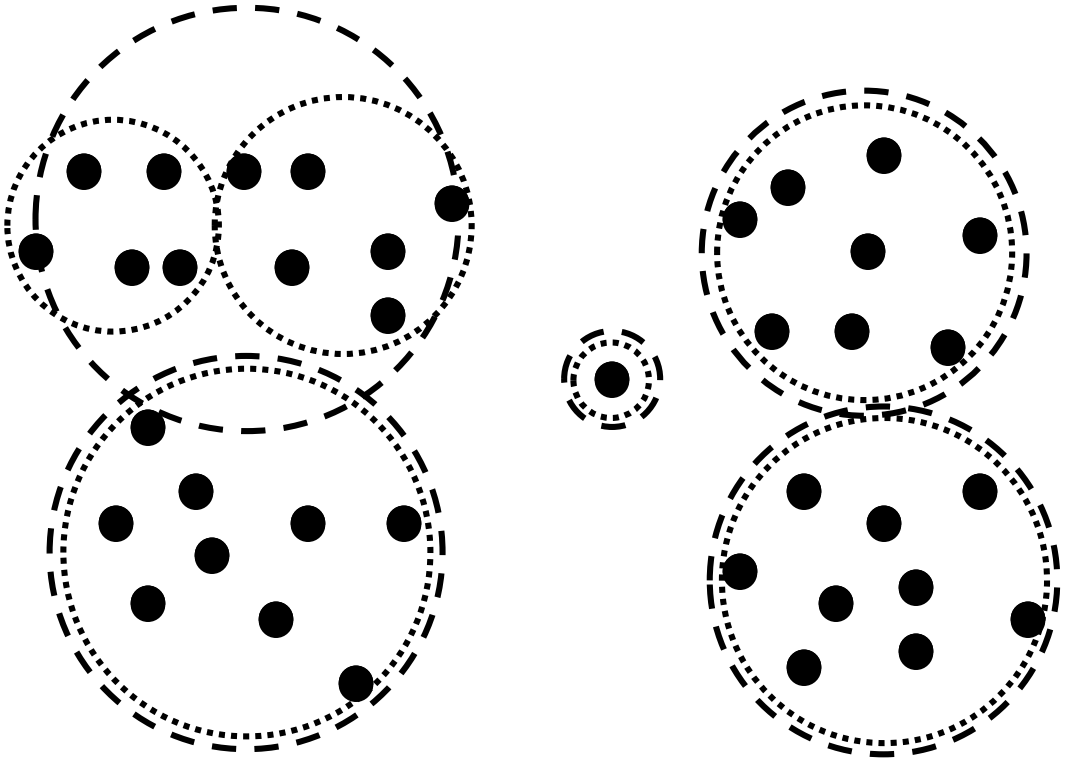
  - Reinicializar las semillas
  - Utilizar técnicas de búsqueda más potentes como algoritmos genéticos o enfriamiento estocástico

Desventajas

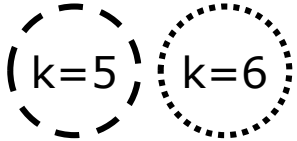
- Sólo es aplicable cuando el concepto de media es definible. ¿Qué hacer con datos nominales?
- Necesidad de fijar anticipadamente el número de *clusters* (k)
- Débil ante datos ruidosos y/o con outliers
- Sólo indicado para *clusters* convexos (esféricos…)

4. Métodos basados en particionamiento. Algunos comentarios sobre k-means

- Necesidad de fijar anticipadamente el número de *clusters* (k)



¿Elección de k?



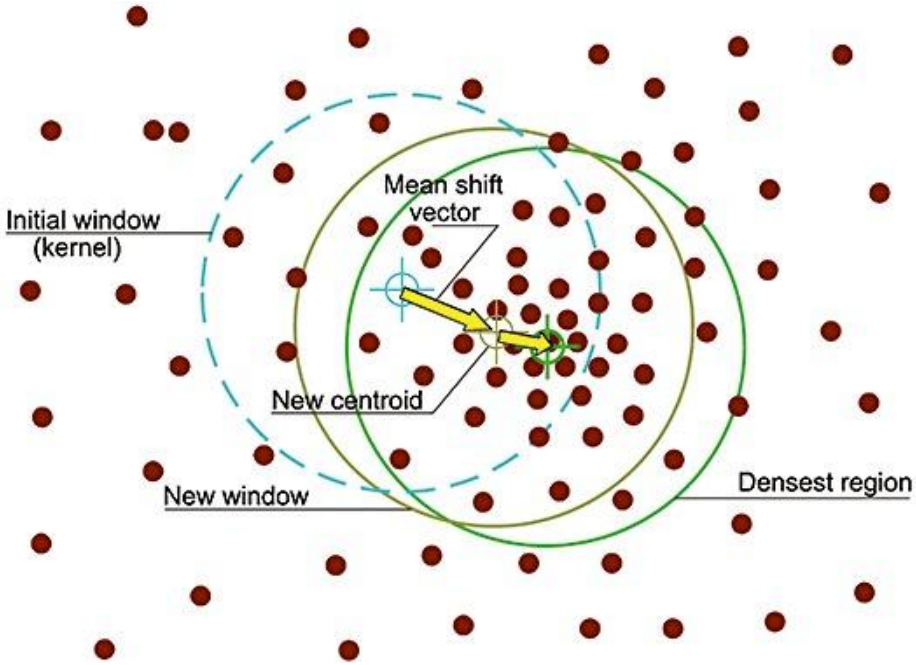
Una opción es iterar con distintos valores de k y elegir la mejor solución en base a alguna medida de rendimiento

4. Métodos basados en particionamiento.
Mean Shift

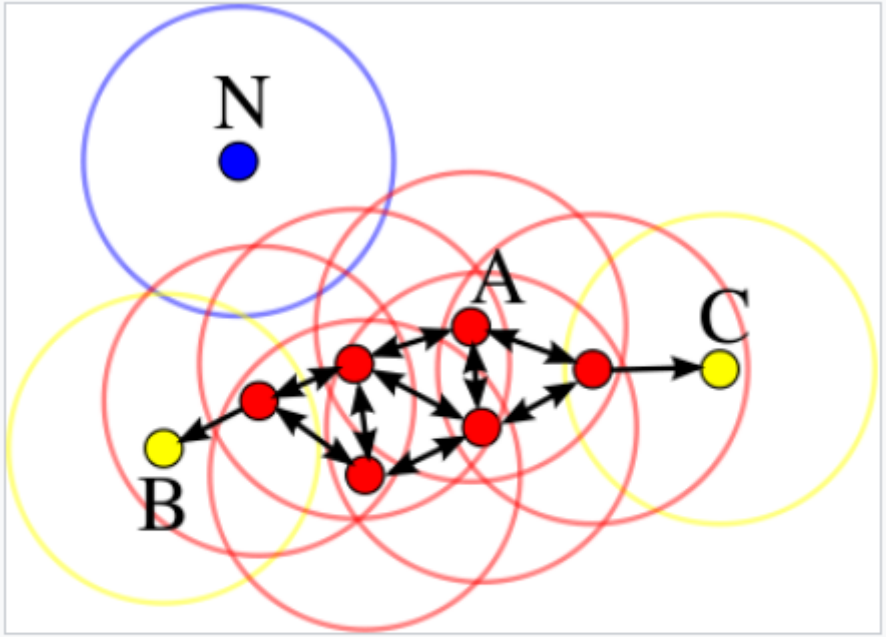
- En lugar de fijar k, el algoritmo MeanShift fija un radio (bandwidth) y va desplazando centroides hasta las regiones más densas

- Se pueden usar kernels gausianos para ponderar los objetos

- El radio se puede estimar con k-NN



4. Métodos basados en particionamiento.
DBSCAN



Los puntos marcados como A son puntos núcleo. Los puntos B y C son *densamente alcanzables* desde A y *densamente conectados* con A, y pertenecen al mismo clúster. El punto N es un punto ruidoso que no es núcleo ni densamente alcanzable. (MinPts=3 o MinPts=4)

https://es.wikipedia.org/wiki/DBSCAN

4. Métodos basados en particionamiento.
DBSCAN

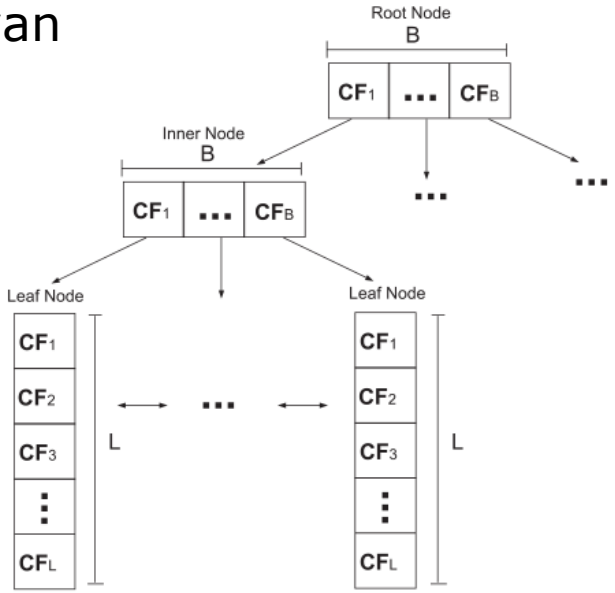
```
DBSCAN(D, eps, MinPts)
   C = 0
   for each unvisited point P in dataset D
      mark P as visited
      NeighborPts = regionQuery(P, eps)
      if sizeof(NeighborPts) < MinPts
         mark P as NOISE
      else
         C = next cluster
         expandCluster(P, NeighborPts, C, eps, MinPts)
expandCluster(P, NeighborPts, C, eps, MinPts)
   add P to cluster C
   for each point P' in NeighborPts
      if P' is not visited
         mark P' as visited
         NeighborPts' = regionQuery(P', eps)
         if sizeof(NeighborPts') >= MinPts
            NeighborPts = NeighborPts joined with NeighborPts'
      if P' is not yet member of any cluster
         add P' to cluster C
regionQuery(P, eps)
   return all points within P's eps-neighborhood (including P)
```

- Parámetros: *eps* (radio) y *minPts* (tamaño mínimo)
- A partir de un punto, va buscando otros puntos en su vecindad y uniéndolos al clúster hasta que no se alcancen más puntos
- *eps* se puede estimar por k-distancia con *k=minPts*
- DBSCAN puede encontrar cluster con distintas formas y es robusto a outliers

https://es.wikipedia.org/wiki/DBSCAN

4. Métodos basados en particionamiento.
BIRCH

- *Clustering* incremental: agrupa conforme se van recibiendo objetos
- Mantiene características de los *clusters* (CF={N,LS,SS}) para resumir los objetos que van llegando



(1) *Incrementality*. A new object $\mathbf{x}^j$ can be easily inserted into CF vector by updating its statistic summaries as follows.

$$LS \leftarrow LS + \mathbf{x}^j$$
$$SS \leftarrow SS + (\mathbf{x}^j)^2$$
$$N \leftarrow N + 1$$

(2) *Additivity*. Two disjoint vectors CF_1 and CF_2 can be easily merged into CF_3 by summing up their components.

$$N_3 = N_1 + N_2$$
$$LS_3 = LS_1 + LS_2$$
$$SS_3 = SS_1 + SS_2$$

$$centroid = \frac{LS}{N}$$

$$radius = \sqrt{\left(\frac{SS}{N} - \left(\frac{LS}{N}\right)^2\right)}$$

$$diameter = \sqrt{\left(\frac{2N * SS - 2 * LS^2}{N(N-1)}\right)}$$

Cada vez que llega un objeto, desciende por el árbol escogiéndose en cada nodo el CF más cercano. Cuando llega a una hoja, si puede ser absorbido por algún CF existente (porque esté en su radio menor que T) se agrega, si no, se crea un CF nuevo si hay menos de L.
Si se alcanza el máximo L, se divide la hoja en dos con el par de CF más lejanos de la hoja anterior.
La agregación y división se propaga recursivamente hacia arriba

4. Métodos basados en particionamiento.
Medidas de rendimiento

- Coeficiente *silhouette*: mide cómo de similares son los objetos de un mismo cluster (cohesión) comparado con otros clusters (separación)

- Sea $a(i)$ la distancia media del objeto i al resto de objetos de su cluster. Sea $b(i)$ la mínima distancia media del objeto i al resto de objetos del resto de clusters

- $s(i)$ toma valores en [-1,1], siendo mejor cuanto más cercano a 1 está. Si $s(i)$ se acerca a cero, significa que el objeto está en el borde de dos clusters

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

- La media de todos los $s(i)$ es el coeficiente *silhoutte* que mide la calidad global del agrupamiento

4. Métodos basados en particionamiento.
Medidas de rendimiento

- Calinski-Harabasz: razón entre la dispersión intra-clusters y la dispersión inter-clusters. Cuanto mayor es el valor, mejor es el agrupamiento

$$\mathrm{CH}(P) = \frac{(N - |P|)\,\mathrm{inter}_{CH}(P)}{(|P| - 1)\,\mathrm{intra}_{CH}(P)}$$

N es el número de objetos

$|P|=k$ es el número de clusters

$$\mathrm{inter}_{CH}(P) = \sum_{C \in P} |C|\,\mathrm{d}(\bar{C}, \bar{X}) \text{ e } \mathrm{intra}_{CH}(P) = \sum_{C \in P} \sum_{x \in C} \mathrm{d}(x, \bar{C})$$

- Otras medidas:

Davies-Bouldin

$$\mathrm{DB}(P) = \frac{1}{|P|} \sum_{C_k \in P} \max_{C_l \in P/C_k} \left\{ \frac{\mathrm{S}(C_k) + \mathrm{S}(C_l)}{\mathrm{d}(\bar{C}_k, \bar{C}_l)} \right\}$$

$$\mathrm{S}(C) = 1/|C| \sum_{x \in C} \mathrm{d}(x, \bar{C}).$$

R-Squared

$$RS = \frac{SS_t - SS_w}{SS_t}$$

$$SS_t = \sum_{j=1}^{d} \sum_{k=1}^{n_j} \left(x_k - \overline{x_j}\right)^2, \; SS_w = \sum_{\substack{i=1\ldots nc \\ j=1\ldots d}} \sum_{k=1}^{n_{ij}} \left(x_k - \overline{x_j}\right)^2$$

Dunn

$$\mathrm{Dunn}(P) = \frac{\mathrm{inter}_{Dunn}(P)}{\mathrm{intra}_{Dunn}(P)}$$

$$\mathrm{inter}_{Dunn}(P) = \min_{C_k \in P} \left\{ \min_{C_l \in P/C_k} \{\delta(C_k, C_l)\} \right\}$$

$$\delta(C_k, C_l) = \min_{x_i \in C_k} \left\{ \min_{x_j \in C_l} \mathrm{d}(x_i, x_j) \right\}$$

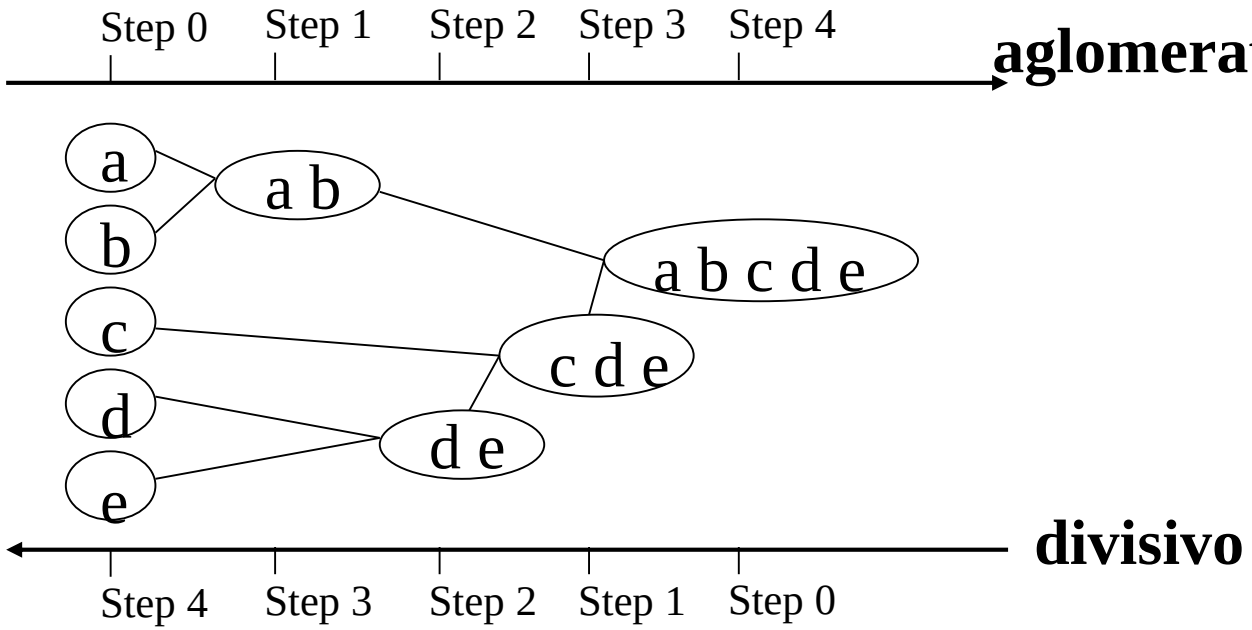
$$\mathrm{intra}_{Dunn}(P) = \max_{C \in P} \left\{ \max_{x_i, x_j \in C} \mathrm{d}(x_i, x_j) \right\}$$

Contenidos sobre Clustering

1. Clustering/agrupamiento/segmentación
2. Medidas de distancia y similaridad
3. Distintas aproximaciones al *clustering*
4. Métodos basados en particionamiento
5. **Métodos jerárquicos**
   - 5.1. Métodos aglomerativos
   - 5.2. Métodos divisivos

5. Métodos jerárquicos

- La salida es una jerarquía entre *clusters*
- Dependiendo del nivel de corte obtendremos un *clustering* distinto
- No requiere como parámetro el número de *clusters*



5.1. Métodos aglomerativos

- Se basan en medir la distancia entre *clusters*
- En cada paso se fusionan los dos *clusters* más cercanos
- La situación de partida suele ser un *cluster* por cada objeto en la BD: $C = \{x_1,...,x_n\}$

- Para $i=1,...,n$ hacer $C_i=\{x_i\}$
- Mientras haya más de un *cluster*
  - Sean C_i y C_j los dos *clusters* que minimizan la distancia entre *clusters*
  - $C_i = C_i \cup C_j$
  - Eliminar el *cluster* C_j

5.1. Métodos aglomerativos

- Minimizar la distancia mínima entre elementos de cada grupo (enlace simple)

$$D_{es}(C_i, C_j) = \min_{i,j}\{d(x,y) \mid x \in C_i, y \in C_j\}$$

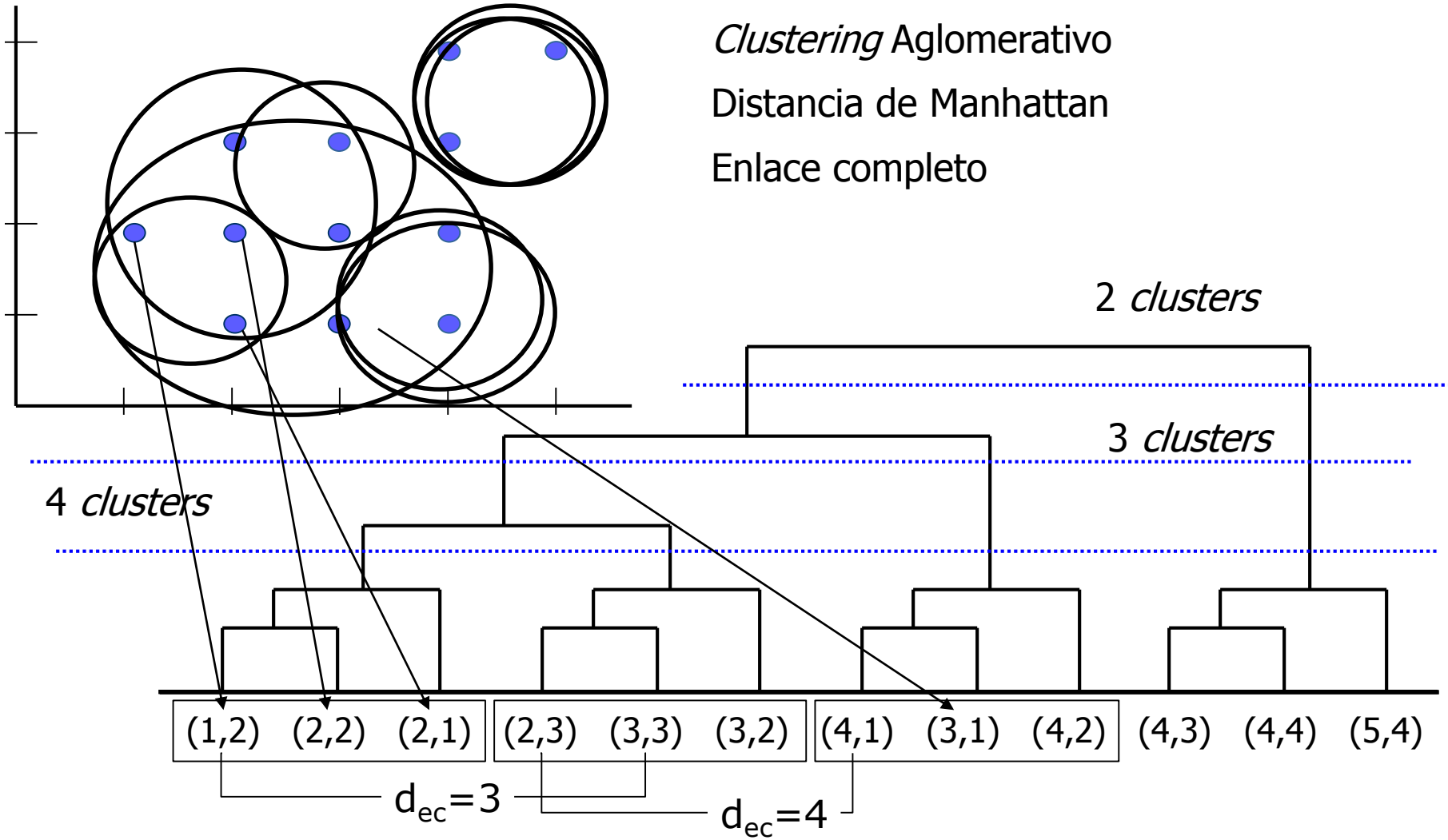
donde *d(,)* es una distancia entre objetos

- Minimizar la distancia máxima entre elementos de cada grupo (enlace completo)

$$D_{ec}(C_i, C_j) = \max_{i,j}\{d(x,y) \mid x \in C_i, y \in C_j\}$$

- Varianza mínima (Ward): fusionar el par de cluster que genera un agrupamiento con mínima varianza (media de la distancia cuadrática de cada elemento al centroide)
- Distancia entre los centroides

5.1. Métodos aglomerativos



5.2. Métodos divisivos

- Comienzan con un único *cluster* (toda la BD) y en cada paso se selecciona un *cluster* y se subdivide
- Se debe dar una condición de parada, o en su defecto se detiene el proceso cuando cada *cluster* contiene un único objeto
- Podemos distinguir dos variantes:
  - Unidimensional (*Monothetic*). Sólo se considera una variable para hacer la partición
  - Multidimensional (*Polythetic*). Todas las variables se consideran para hacer la partición
    - Se usa una distancia entre *clusters* para medir
- Mucho menos utilizados que los métodos aglomerativos con un bajo número de vecinos "cercanos"

INTELIGENCIA DE NEGOCIO

2018 - 2019

- Tema 1. Introducción a la Inteligencia de Negocio
- Tema 2. Minería de Datos. Ciencia de Datos
- Tema 3. Modelos de Predicción: Clasificación, regresión y series temporales
- Tema 4. Preparación de Datos
- Tema 5. Modelos de Agrupamiento o Segmentación
- Tema 6. Modelos de Asociación
- Tema 7. Modelos Avanzados de Minería de Datos.
- Tema 8. Big Data